

Sprint Evaluation

What is the progress of your project in this sprint?

In this sprint, progress of developing our application keeps continuing as it was planned. We dealt with different parts of the project and we overcame different hard tasks while developing application. In the beginning of each week of the Sprint 2, each one of us had researched to overcome the tasks which we had done. Before starting sprint 2, we have already specified our tasks in our start-up document. The goals which were to be achieved up to sprint 2 are done in our schedule. In addition to our specified tasks, we have done additional tasks like database designation. In the end of the sprint 2, we were able to see the overall data structure of our project so that we have created Entity Relationship and Class Diagram. We have showed static view of our application in our class diagram and database relations on E-R diagram. In the first retrospective report, we have specified that we may revise the tasks on the sprint 2 and maybe we could do some of tasks which are told to be done in sprint 3. For example, database revision with respect to phone numbers which is told to be done in sprint 3 has been done already in sprint 2. In addition, in the first retrospective report we had stated that using sqlite which is androids' self database is redundant. However, in this sprint 2 we have realized that, when we fetch all data from our server database the application dramatically slows down. Instead of this approach, in this sprint we have decided to use sqlite to store instant data which application consumes. By doing so, we will not need every data in the server database, we will use server database only when user updates the information in the application. Furthermore, we have integrated Google Cloud Messaging which is a third party cloud for chat activity. GCM is a free service which mediates between server database and android clients to send and get messages.

What goals are achieved? What problems are overcome?

We have encountered different kinds of problems in our progress and many of them are overcome by now. In improvement stage, later on, we will encounter so many problems. Up to now, the problems we have encountered were to create Android List Views and Layouts, verifying the phone number verification, scaling photos without reducing the resolution too much. First problem is the creating a list view to place the all registered users by fetching the users from our database. In the beginning of this task, fetching those data and placing on the Listview was not synchronized. Before fetching data we were trying to show those data in the listview, we have overcome this problem by synchronizing the fetch process and show process. Moreover, phone number verification was really exhaustive task because to verify the phone number in real time was handled with SDK named SYNC. If we use this SDK, the size of the application would increase dramatically and using this SDK is non-free. After recognizing this problem, we decided to use SMS system. Phone number verification was done with SMS in the following scenario. To verify the phone number, Android send a SMS message to user's phone with the phone number of that user. We checked on the Android Side if the phone number in the sended message equals to that user's phone then phone number verified. This is our solution for verifying the phone number. In addition to those problems, scaling the user profile photos without resolution was exhaustive task because there is not any library for reducing those photos without reducing the resolution of photos. The solution for this problem is to scale the bitmaps of the photos and send those bitmaps byte by byte. This is the technique like Whatsapp. Those are our problem for this sprint.

Up to the end of Sprint2,

- Registration with phone number was achieved.
- Sending and accepting friend requests was completed and also tables for achieving friend relations was created.
- Integration Google Cloud Messaging registration and phone number verification system has been done.
- User profile pictures scaled by 80 like Whatsapp without reducing the resolution too much on the Android Side for minimizing the time while uploading to our server. After scaling on Android Side, sending photos to the server has been achieved. For Instance, time for uploading a photo with size 2MB is about 1 sec.

- Creating GCM structure and integrating that structure to our system has been done. Moreover, registering the user to that system with unique registration ID has been achieved.
- Creating chat interface with chat bubbles are achieved. Received and sended messages are shown in this Interface.
- Message flowing between registered users is the main part of Google Cloud Messaging. Sending message is done with the relation between PHP and Android Java Files. Furthermore, capturing sended messages on the Android side is done with the helper Java class such as broadcast receiver.

If you are updating your plans what are your justifications?

Up to now, we have achieved the tasks in sprint 1 and sprint 2. Although there are small changes in our plans like deciding to use sqlite again, we haven't changed our overall structure of our application. Justification for using sqlite again is described in the progress part. Apart from that, there is no dramatic change.

Team evaluation

How well your is team working together? How many meetings did you hold? Are you planning any changes in your cooperation strategy? Which work is completed by which member (in a Gannt chart)?

For the second sprint, we did not change the way that we work. We have been handling our obligations by selecting appropriate tasks for ourselves. We continue our meeting with our assistant on thursday 4:00 pm as scheduled. With our supervisor we have set the meeting time on thursday 2:00 PM in every week.

We have learnt using gitlab in sprint 2 and we have pushed our master project to gitlab. Before that, we had to integrate the project by hand. This approach developed productivity of our team working. Since our project is an idea project, we are not planning to change in cooperation strategy.

Task	Assigned Member	1 st week	2 nd week	3 rd week
Phone Number Verification	Mehmet Gençol	Research about number verification systems.	Testing external SDK and SMS verification. Chasing SMS verification.	SMS verification without SDK and integration to master.
Database Revision& Making Phone Number Unique	Mehmet Gençol	Research about how to revise database effectively	Revising database while adding phone number.	Making phone number unique.

Merge view/activities	Mehmet Gençol	Creation of views(by all members according to our responsibility)	Decrementing the number of activities.	Inserting views to NERS.
User Relations	Ahmet Melih Gedikli	Add Friend and Waiting requests activity was created. Users are able to send, and accept friend requests and see them on listview.	After sending and accepting friend requests, it is in-service to see friends for users on showlistview activity.	In this period, online/offline status of friends and users are also available.
Integrating GCM registration and phone verification	Ahmet Melih Gedikli	Research about Google Cloud Messaging(GCM)	Setting up phone verification.	In registration process, it is obligatory to verify phone numbers for users before GCM registration.
Directing ShowFriend List items to GCM chat activity	Ahmet Melih Gedikli	Research about layouts that will be used for ShowFriendList items.	Layouts for ShowFriendList items are created.	It is in-service to be redirected to chat activity for users when they click show friendlist items. One to one chat is set in this way.
Scaling and Sending User Profile Images to Server	Mustafa Murat Coşkun	Research about techniques of scaling pictures.	Scaling user profile picture on Android Side by 80 like Whatsapp Technique.	Sending image to server using Volley Library
Creating Chat Interface	Mustafa Murat Coşkun	Research about layouts that will be used for chat interface.	Creating simple chat interface was handled in this week.	Received and sent Messages are shown in this chat interface.
Creating simple chat system and message flow between registered users	Mustafa Murat Coşkun	Using GCM system, message flowing between users was also handle in this week.To send message to a specific user, relation between PHP and Android files were created.	To capture the sended messages, BroadcastReceiver was used on the Android Side.	To capture the sended messages, LocalBroadCast Receiver was used on the Android Side.
Notification System	Oğuz Artıran	Research about notifications and services in Android.	Notifications that will inform users when a friend of user login to the application are	Notifications that will inform users when a friend of user login to the application are created while

			created while application is open.	application is close.
Update on information which database contains	Oğuz Artıran	Research about how to hold the online status of users into the database updated.OnCreate, onPause and onDestroy overridden methods are researched	When a user login to the system, his/her status are updated into the database with making his/her status value 1.	When a user logout from the system, his/her status are updated into the database with making his/her status value 0.
Other messaging SDKs that we may use	Oğuz Artıran	Research about other messaging systems like Parse and Sinch.Parse is tried because it is seen simple compare to the others.	Registration into the Parse system.Dashboard of Parse is used and Parse is integrated to the application.	Simple chat system is created.Users can send message to each other and these messages are recorded into the dashboard of Parse.
Backlog Updates				
<p>What are your backlog updates?</p> <p>When creating start-up document, we did not expected to add phone numbers into database. In addition, we have implemented some features from sprint 2. This creates a need for us to implement some tasks from sprint 3. We have added phone number and should make unique identifier as phone number. However, GCM system requires incrementing id. Hence, we have marked phone number as unique and picked unique identifier as id which is incremented automatically.</p>				